UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/610,301	07/05/2000	Derek W. Bolton	CSC01275	4206

29989 7590 07/01/2004

HICKMAN PALERMO TRUONG & BECKER, LLP
1600 WILLOW STREET
SAN JOSE, CA 95125

EXAMINER	
KADING, JOSHUA A	

ART UNIT	PAPER NUMBER
2661	10

DATE MAILED: 07/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/610,301	BOLTON, DEREK W.	
	Examiner	Art Unit	
	Joshua Kading	2661	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 May 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19,21,23-33,35 and 36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19,21,23-33,35 and 36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 July 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

Drawings

This application, filed under former 37 CFR 1.60, lacks formal drawings. The informal drawings filed in this application are acceptable for examination purposes. When the application is allowed, applicant will be required to submit new formal drawings. In unusual circumstances, the formal drawings from the abandoned parent application may be transferred by the grant of a petition under 37 CFR 1.182.

Claim Rejections - 35 USC § 103

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claims 1, 3-8, 11-16, 19, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over RFC2582

Regarding claim 1, RFC2582 discloses "a network device-based method comprising:

determining...upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements, wherein the excess number of duplicate acknowledgements is a number that represents an amount of duplicate acknowledgements and is based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 whereby receiving 3 duplicate ACK's to engage the recovery means that an excess number of duplicate ACK's was determined, and that determination triggers the start of the recovery process, further, although there is no mention of "receipt of new data" in section 3, it is well known in the art (especially in ACK type systems that the first ACK received of duplicates is in response to new data); and

taking a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 explicitly lacks "...retaining..." the excess number of duplicate ACKs as they are received. Although RFC2582 does not explicitly disclose "...retaining..." the excess number of duplicate ACKs as they are received, it would have obvious to one with ordinary skill in the art at the time of invention to have the excess number of duplicate ACKs retained for the purpose comparing this count to the threshold to determine if the recovery process should be engaged. The motivation is that a count of duplicate ACKs must be retained so it can, at a future time, be compared with a

threshold. If the count is not stored then the threshold will never be reached and the recovery process will never engage.

Regarding claim 12, RFC2582 discloses "a network device-based method comprising:

determining...upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements, wherein the excess number of duplicate acknowledgements is a number that represents an amount of duplicate acknowledgements and is based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 whereby receiving 3 duplicate ACK's to engage the recovery means that an excess number of duplicate ACK's was determined, and that determination triggers the start of the recovery process, further, although there is no mention of "receipt of new data" in section 3, it is well known in the art (especially in ACK type systems that the first ACK received of duplicates is in response to new data);

deflating a congestion window upon said value of excess number duplicate acknowledgements being less than a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 7-13); and

optimizing a size of said congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being

greater than a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 24-31)."

RFC2582 explicitly lacks "...retaining..." the duplicate ACKs as they are received. Although RFC2582 does not explicitly disclose "...retaining..." the duplicate ACKs as they are received, it would have obvious to one with ordinary skill in the art at the time of invention to have the excess number of duplicate ACKs retained for the purpose comparing this count to the threshold to determine if the recovery process should be engaged. The motivation is that a count of duplicate ACKs must be retained so it can, at a future time, be compared with a threshold. If the count is not stored then the threshold will never be reached and the recovery process will never engage.

Regarding claims 3 and 13, RFC2582 discloses the methods of claims 1 and 12. RFC2582 further discloses "deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 7-13)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the deflating of a congestion window with the methods of claims 1 and 12 for the same reasons and motivation as in claims 1 and 12.

Regarding claim 4, RFC2582 discloses the method of claim 1. RFC2582 further discloses "optimizing a size of a congestion window to match a reduction in a quantity of

unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 24-31)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the optimizing of a congestion window with the method of claim 1 for the same reasons and motivation as in claim 1.

Regarding claim 5, RFC2582 discloses the method of claim 1. RFC2582 further discloses "comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where it is suggested that the value of 3 is the threshold to which the count is being compared to)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the comparing the excess number of duplicate acknowledgements with the method of claim 1 for the same reasons and motivation as in claim 1.

Regarding claims 6 and 14, RFC2582 discloses the methods of claims 5 and 13. RFC2582 further discloses "performing a fast retransmit upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 3-9 where the Fast Retransmit is part of the recovery method). It would have been obvious to one with

ordinary skill in the art at the time of invention to include the performing a fast retransmit with the methods of claims 5 and 13 for the same reasons and motivation as in claims 5 and 13.

Regarding claims 7 and 15, RFC2582 discloses the methods of claims 6 and 14. RFC2582 further discloses "analyzing a size of a congestion window (section 4 "Resetting the Retransmit Timer", lines 15-17 of section 4 where it is implied that the window is analyzed for size to know how many data packets to transmit)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing a size of a congestion window with the methods of claims 6 and 14 for the same reasons and motivation as in claims 6 and 14.

Regarding claims 8 and 16, RFC2582 discloses the methods of claims 7 and 15. RFC2582 further discloses "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size (section 5 "Avoiding Multiple Fast Retransmits, line 14 of section 5 where it is implied the congestion window is reduced after being analyzed)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the resizing of said congestion window with the methods of claims 7 and 15 for the same reasons and motivation as in claims 7 and 15.

Regarding claims 11 and 19, RFC2582 discloses the methods of claims 1 and 12. RFC2582 further discloses "said method is included in Transmission Control Protocol congestion avoidance (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 1-3 of section 3 where it is the purpose of the fast recovery algorithm to avoid congestion)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the TCP congestion avoidance with the methods of claims 1 and 12 for the same reasons and motivation as in claims 1 and 12.

Regarding claim 33, RFC2582 discloses "a network device-based method for determining...upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements, wherein the excess number of duplicate acknowledgements is a number that represents an amount of duplicate acknowledgements and is based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 whereby receiving 3 duplicate ACK's to engage the recovery means that an excess number of duplicate ACK's was determined, and that determination triggers the start of the recovery process, further, although there is no mention of "receipt of new data" in section 3, it is well known in the art (especially in ACK type systems that the first ACK received of duplicates is in response to new data); and

taking a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 explicitly lacks "...retaining..." the duplicate ACKs as they are received. Although RFC2582 explicitly lacks "...retaining..." the duplicate ACKs as they are received, it would have obvious to one with ordinary skill in the art at the time of invention to have the excess number of duplicate ACKs retained for the purpose comparing this count to the threshold to determine if the recovery process should be engaged. The motivation is that a count of duplicate ACKs must be retained so it can, at a future time, be compared with a threshold. If the count is not stored then the threshold will never be reached and the recovery process will never engage.

Both RFC2582 and Lakshman lack "a programmable memory including a fast recovery extended method..." Although both RFC2582 and Lakshman explicitly lack a programmable memory storing the method, it would have been obvious to one with ordinary skill in the art at the time of invention to include the programmable memory with the method on it because this is the most efficient and feasible way to implement a method in a computer based communication system and since the method steps cannot exist outside of a medium, there must be some type of programmable memory to write and store the method onto.

Claims 2, 9-10, 17-18, 21, 23-32, and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over RFC2582 in view of Chapman et al. (U.S. Patent 6,493,316 B1)

Regarding claims 9 and 17, RFC2582 discloses the methods of claims 1 and 12. RFC2582 lacks "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold." However, Chapman discloses "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold (col. 5, lines 33-34 where inflating the window after the receipt of a non-duplicate ACK is received is a method of determining if the window is inflated and this situation occurs when the duplicate acknowledgement threshold is not met)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing of the congestion window with the methods of claims 1 and 12 for the purpose of controlling flow of packets into the network. The motivation being that by controlling flow into the network congestion and packet loss are reduced to a minimum or tolerable level (Chapman, col. 3, lines 16-30).

Regarding claims 10 and 18, RFC2582 and Chapman disclose the methods of claims 9 and 17. RFC2582 lacks "resizing said congestion window upon analyzing said size of said congestion window showing said size is greater than a predefined size." However, Chapman et al. disclose "resizing said congestion window upon analyzing said size of said congestion window showing said size is greater than a predefined size (col. 8, lines 2-5 where MAX-WND is the predetermined size and C-WND is the actual size of the window)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the resizing of the congestion window with the methods of claim 9 and 17 for the same reasons and motivation as in claims 9 and 17.

Regarding claim 21, RFC2582 discloses "...a fast recovery extended method (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5 where retransmitting the partial segment in response to the partial ACK is an extended packet transmission recovery action)...determine, upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements, wherein the excess number of duplicate acknowledgements is a number that represents an amount of duplicate acknowledgements and is based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 whereby receiving 3 duplicate ACK's to engage the recovery means that an excess number of duplicate ACK's was determined, and that determination triggers the start of the recovery process, further, although there is no mention of "receipt of new data" in section 3, it is well known in the art (especially

in ACK type systems that the first ACK received of duplicates is in response to new data)...and take a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 lacks "a processor; and a memory coupled to said processor, and storing a fast recovery extended method wherein upon execution of said fast recovery extended method by said processor..." However, Chapman disclose "a processor; and a memory coupled to said processor, and storing a fast recovery extended method wherein upon execution of said fast recovery extended method by said processor a fast recovery process is extended (figure 15, elements 32, 30, and 28; col. 8, lines 27-31 where elements 32 and 28 constitute the processor and 30 is the memory that stores the fast recovery extended method; although the fast recovery extended method is not explicitly disclosed in Chapman, it is suggested that by using the TCP methods of RFC2582 with Chapman, they would need to be stored in the memory in order to be executed)..."

RFC2582 and Chapman further lack to "...retain said excess number of duplicate acknowledgements in said memory..." Although RFC2582 does not explicitly mention retaining of "said excess number of duplicate acknowledgements in said memory, it would have obvious to one with ordinary skill in the art at the time of invention to have the excess number of duplicate ACKs retained for the purpose comparing this count to the threshold to determine if the recovery process should be engaged. The motivation is

that a count of duplicate ACKs must be retained so it can, at a future time, be compared with a threshold. If the count is not stored then the threshold will never be reached and the recovery process will never engage.

Regarding claims 2 and 23, RFC2582 disclose the method of claim 1; and RFC2582 and Chapman disclose the method of claim 21. RFC2582 lacks "determining whether a congestion window is inflated prior to said determining an excess number of duplicate acknowledgements." However, Chapman discloses "determining whether a congestion window is inflated prior to said determining an excess number of duplicate acknowledgements (col. 5, lines 33-34 where inflating the window after the receipt of a non-duplicate ACK is an indicator that the window is inflated and thus all that is needed to do to determine if the window is inflated is look to see if the last ACK is a non-duplicate). It would have been obvious to one with ordinary skill in the art at the time of invention to include the determining whether a congestion window is inflated with the methods of claims 1 and 21 for the purpose controlling flow of packets into the network. The motivation being that by controlling flow into the network congestion and packet loss are reduced to a minimum or tolerable level (Chapman, col. 3, lines 16-30).

Regarding claim 24, RFC2582 and Chapman disclose the method of claim 21. Chapman lack "deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment." However, RFC2582 further discloses

"deflating a congestion window upon said value of said excess number of duplicate acknowledgements in bytes being less than a number of bytes in a transmission control protocol sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 7-13)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the deflating of a congestion window with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 25, RFC2582 and Chapman disclose the method of claim 21. Chapman lack "optimizing a size of a congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment." However, RFC2582 further discloses "optimizing a size of a congestion window to match a reduction in a quantity of unacknowledged data upon said excess number of duplicate acknowledgements being greater than a TCP sender segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", step 5, lines 24-31)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the optimizing of a congestion window with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 26, RFC2582 and Chapman disclose the method of claim 21. Chapman lack "comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold." However, RFC2582 further discloses

"comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 where it is suggested that the value of 3 is the threshold to which the count is being compared to)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the comparing the excess number of duplicate acknowledgements with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 27, RFC2582 and Chapman disclose the method of claim 26. Chapman lack "performing a fast retransmit upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold." However, RFC2582 further discloses "performing a fast retransmit upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is greater than or equal to said duplicate acknowledgement threshold (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 3-9 where the Fast Retransmit is part of the recovery method). It would have been obvious to one with ordinary skill in the art at the time of invention to include the performing a fast retransmit with the method of claim 26 for the same reasons and motivation as in claim 26.

Regarding claim 28, RFC2582 and Chapman disclose the method of claim 27. Chapman lack "analyzing a size of a congestion window." However, RFC2582 further discloses "analyzing a size of a congestion window (section 4 "Resetting the Retransmit Timer", lines 15-17 of section 4 where it is implied that the window is analyzed for size to know how many data packets to transmit)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing a size of a congestion window with the method of claim 27 for the same reasons and motivation as in claim 27.

Regarding claim 30, RFC2582 and Chapman disclose the method of claim 21. RFC2582 lacks "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold." However, Chapman discloses "analyzing a size of a congestion window upon said comparing said excess number of duplicate acknowledgements with a duplicate acknowledgement threshold indicating that said excess number of duplicate acknowledgements is less than said duplicate acknowledgement threshold (col. 5, lines 33-34 where inflating the window after the receipt of a non-duplicate ACK is received is a method of determining if the window is inflated and this situation occurs when the duplicate acknowledgement threshold is not met)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the analyzing of the congestion window with the method of claim 21

for the purpose of controlling flow of packets into the network. The motivation being that by controlling flow into the network congestion and packet loss are reduced to a minimum or tolerable level (Chapman, col. 3, lines 16-30).

Regarding claims 29 and 31, RFC2582 and Chapman disclose the methods of claims 28 and 30. Chapman lacks "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size." However, RFC2582 further discloses "resizing said congestion window upon said analyzing said size of said congestion window showing said size is greater than a predefined size (section 5 "Avoiding Multiple Fast Retransmits, line 14 of section 5 where it is implied the congestion window is reduced after being analyzed)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the resizing of said congestion window with the methods of claims 28 and 30 for the same reasons and motivation as in claims 28 and 30.

Regarding claim 32, RFC2582 and Chapman disclose the method of claim 21. Chapman lack "said method is included in Transmission Control Protocol congestion avoidance." However, RFC2582 further discloses "said method is included in Transmission Control Protocol congestion avoidance (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 1-3 of section 3 where it is the purpose of the fast recovery algorithm to avoid congestion)." It would have been obvious to one with ordinary skill in the art at the time of invention to include the TCP

congestion avoidance with the method of claim 21 for the same reasons and motivation as in claim 21.

Regarding claim 35, RFC2582 "determining, upon receiving acknowledgement of receipt of new data, an excess number of duplicate acknowledgements, wherein the excess number of duplicate acknowledgements is a number that represents an amount of duplicate acknowledgements and is based upon a count of consecutive duplicate acknowledgement packets (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 whereby receiving 3 duplicate ACK's to engage the recovery means that an excess number of duplicate ACK's was determined, and that determination triggers the start of the recovery process, further, although there is no mention of "receipt of new data" in section 3, it is well known in the art (especially in ACK type systems that the first ACK received of duplicates is in response to new data); and

taking a network packet transmission recovery action based upon said excess number of duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 8-13 of section 3 when a threshold of duplicate acknowledgements is reached the "Fast Recovery procedure" begins)."

RFC2582 lacks means for determining and means for taking recovery action. However, Chapman further discloses means for determining and means for taking recovery action (figure 15 which is a network device that implements all the above tasks).

RFC2582 and Chapman further lack "...retaining..." the duplicate ACKs as they are received. Although RFC2582 explicitly lacks "...retaining..." the duplicate ACKs as they are received, it would have obvious to one with ordinary skill in the art at the time of invention to have the excess number of duplicate ACKs retained for the purpose comparing this count to the threshold to determine if the recovery process should be engaged. The motivation is that a count of duplicate ACKs must be retained so it can, at a future time, be compared with a threshold. If the count is not stored then the threshold will never be reached and the recovery process will never engage.

Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over RFC2582 in view of Lakshman et al. (U.S. Patent 6,078,564).

Regarding claim 36, RFC2582 discloses "a method for recovery of multiple transmission units comprising:

setting a duplicate acknowledgements threshold, wherein a duplicate acknowledgement is an acknowledgement of receipt of a transmission unit for which an acknowledgement already exists (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16 where it is implied that because the recovery process is engaged when 3 duplicate ACKs are received, there must be a duplicate ACK threshold and it must have been set at some point);

setting a size for a congestion window (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", line 20);

determining a value representing a count of consecutive duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16 where the 3 duplicate ACKs received is a count);

if the value is equal to the duplicate acknowledgement threshold, performing a first fast retransmit operation in which at least one of the transmission units is retransmitted (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16 and line 20 shows the retransmit), and resizing the size of the congestion window (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", line 20);

determining whether any subsequent duplicate acknowledgements were received (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 24-26);

in response to receipt of each of the subsequent duplicate acknowledgements, increasing the size of the congestion window (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 24-26), and if transmitting another segment is permitted, transmitting another segment (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 27-28);

and when an acknowledgement for the transmission unit that was retransmitted is received, performing a fast recovery including at least a get excess operation which at least determines a value representing an excess number of duplicate acknowledgements based upon the value of the count of consecutive duplicate acknowledgements for the retransmitted transmission units (section 3 "The Fast

Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16 where the excess duplicate ACKs are what triggers the recovery procedure and since the trigger is a threshold, the duplicate ACKs must be counted so that they may be compared against the threshold),

a recovery action operation, in which at least the sender initiates one or more network packet transmission recovery actions based upon the excess duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16), wherein the network packet transmission recovery actions include at least taking no further action (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16 where no action is taken if the threshold (3 duplicate ACKs) is not met), deflating the size of the congestion window (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 37-40), resizing the size of the congestion window to a more optimal size (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", line 20 where setting the window to this value allows the window to be as big as the buffered segments), performing another fast retransmit (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", line 20), resizing the size of the congestion window from the more optimal size (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 24-26), and resizing the size congestion window after the deflating (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 52-55), and a set duplicate acknowledgment operation in which at least the value representing the count of the duplicate acknowledgements is set equal to the value

representing the excess duplicate acknowledgements (section 3 "The Fast Retransmit and Fast Recovery Algorithms in NewReno", lines 13-16 where, for instance, if the threshold is zero, the count is equal to the excess)."

However, RFC2582 lacks what Lakshman discloses, "...transmitting a plurality of transmission units from a sender to a receiver (figure 1, elements 15, 16, and B_f where the buffer has a plurality of transmission units and 15 is the transmitter and 16 is the receiver), wherein the receiver is an entity that is currently receiving transmission units, and wherein the sender is an entity that is currently sending the transmission units (figure 1, elements 15 and 16); the receiver transmitting acknowledgements of receipt of the transmission units received (figure 1, element B_r is a buffer of ACKs)..."

It would have been obvious to one with ordinary skill in the art at the time of invention to include the sender, receiver, and ACKs with the rest of the method for the purpose of communicating via data packets with one another (Lakshman, col. 2, lines 14-19). The motivation is that communication using data packets is fast and reliable.

Response to Arguments

The 35 U.S.C. 112 first paragraph rejection of the previous Office Action has been withdrawn in light of applicant's amendments and further explanation.

Applicant's arguments filed 18 May 2004 have been fully considered but they are not persuasive.

Applicant argues that RFC2582 does not show determining a number of excess duplicate acknowledgements. The examiner respectfully disagrees. As discussed in the above rejections, RFC2582 suggests that an excess number of duplicate acknowledgements is determined from the received acknowledgements. As read in RFC2582, section 3, the recovery procedure begins when 3 duplicate acknowledgements are received. This implies a threshold of acknowledgements has been reached and any more duplicate ACKs received are in excess. The act of determining is the implied act of comparing the number of duplicate acknowledgements with the threshold to see if there are excess. In RFC2582 for example, the threshold could be 2 and therefore the third duplicate ACK would be in excess causing the recover procedure to begin.

Applicant argues that RFC2582 does not disclose or suggest the retaining of the excess number of duplicate ACKs. The examiner respectfully disagrees. As noted in the above rejections, RFC2582 strongly implies that a count (including the excess) of duplicate acknowledgements is stored and then used to determine if a recovery process should be entered into or not. It is well accepted in the art, that to compare two values electronically, each must be stored. In the case of RFC2582, there is an unmentioned threshold that is used to compare against a number (a count) of received duplicate acknowledgements. These two values must be retained (stored) for later comparison.

Applicant further argues that RFC2582 discloses the duplicate acknowledgements but does not disclose receiving them after "receipt of new data", and that these duplicate ACKs are not received consecutively. The examiner again respectfully disagrees. In an ACK type system, the ACK is sent in response to received data at the receiving end of a system. This is done to tell the transmitter that the data was received successfully and transmission of new data may begin. This process alone signifies a receipt of new data, as each data transmission is unique and thus each first ACK must also be unique. And since a set of 3 duplicate ACKs must begin with a first ACK, it logically follows that this first ACK was in response to new data. If it weren't in response to new data, than it would be a duplicate ACK of a previous transmission, and thus closer to reaching a duplicate ACK threshold.

Accordingly, if an ACK is received that is not the same as the previous (i.e. not a duplicate ACK), it is assumed that there is no problem and no recovery process is engaged and the threshold is set to zero, thus implying that the ACKs are received consecutively. Applicant acknowledged this point in his previous Remarks filed 5 January 2004, page 13, lines 6-9.

Applicant argues that the variables "recover" and "send_high" are not used to stored the number of excess number of duplicate acknowledgements. The examiner agrees but is confused as to why these are being argued in the first place? Nowhere in the previous Office Actions are the variables "recover" and "send_high" used to describe

the storing of the number of excess duplicate ACKs. As in the current office action, the retaining of the number of excess duplicate ACKs is implied in RFC2582.

Given the current rejections and further explanations in this section, rejections for the dependent claims are maintained.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua Kading whose telephone number is (703) 305-0342. The examiner can normally be reached on M-F: 8:30AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Douglas Olms can be reached on (703) 305-4703. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Joshua Kading
Examiner
Art Unit 2661

June 22, 2004

KENNETH VANDERPUYE
PRIMARY EXAMINER